Constraint Satisfaction

CSE 473
University of Washington

Logistics

- PS
  PS 1 due in a week
- Reading
  Today + Monday: Ch 5 (esp 5.1 and 5.2)
  4/14 (Wed) Ch 6 (esp 6.1 - 6.4)
  4/16 (Fri) Ch 7

473 Topics

Blind
Informed (Heuristics)
Constraint Satisfaction (Factored)
Adversarial

Search
Knowledge
Representation

Problem Spaces
Agency

Today: Constraint Satisfaction Problems

- Definition
  Factoring state spaces
- Backtracking policies
- Variable-ordering heuristics
- Preprocessing algorithms

Constraint Satisfaction

- Kind of search in which
  States are factored into sets of variables
  Search = assigning values to these variables
  Structure of space is encoded with constraints
- Backtracking-style algorithms work
  E.g. DFS for SAT (i.e. DPLL)
- But other techniques add speed
  Propagation
  Variable ordering
  Preprocessing

Chinese Food as Search?

- States?
  - Partially specified meals
- Operators?
  - Add, remove, change dishes
- Start state?
  - Null meal
- Goal states?
  - Meal meeting certain conditions (rating?)
Factoring States
• Rather than state = meal
• Model state’s (independent) parts, e.g.
  Suppose every meal for n people
  Has n dishes plus soup
  Soup
  Meal 1
  Meal 2
  Meal n
• Or... physical state =
  X coordinate =
  Y coordinate =

CSPs in the Real World
• Scheduling space shuttle repair
• Airport gate assignments
• Transportation Planning
• Supply-chain management
• Computer configuration
• Diagnosis
• UI optimization
• Etc...

Binary Constraint Network
Partial assignment of values = tuple of pairs
  (x₁, a₁) ... (xₙ, aₙ) means variable x gets value a...
Tupled = consistent if all constraints satisfied
Tupled = full solution if consistent + has all vars

Tupled = \{ (x₁, a₁) ... (xₙ, aₙ) \} = consistent w/ a set of vars
  \{ x_m ... x_n \}
iff \exists a_m ... a_n such that
  \{ (x₁, a₁) ... (xₙ, aₙ), (x_m, a_m) ... (x_n, a_n) \} = consistent

Chinese Constraint Network

Binary Constraint Network
• Set of n variables: x₁ ... xₙ
• Value domains for each variable: D₁ ... Dₙ
• Set of binary constraints (also "relations")
  Rᵢ ⊆ Dᵢ × Dᵢ
  Specifies which values pair (xᵢ, xⱼ) are consistent
• V for each country
• Each domain = 4 colors
• Rᵢ enforces ≠

Cryptarithmetic
• State Space
  Set of states
  Operators [and costs]
  Start state
  Goal states
• Variables?
• Domains (variable values)?
• Constraints?
Classroom Scheduling

- Variables?
- Domains (possible values for variables)?
- Constraints?

N Queens

- As a CSP?

N Queens

- Variables = board columns
- Domain values = rows
- \( R_{ij} = \{(a_i, a_j) : (a_i \neq a_j) \land (|i-j| \neq |a_i-a_j|)\} \)
  e.g. \( R_{12} = \{(1,3), (1,4), (2,4), (3,1), (4,1), (4,2)\} \)

\( \{(x_1, 2), (x_2, 4), (x_3, 1)\} \) consistent with \( x_4 \)
- Shorthand: "\( \{2, 4, 1\} \) consistent with \( x_4 \)"

CSP as a search problem?

- What are states?
  (nodes in graph)
- What are the operators?
  (arcs between nodes)
- Initial state?
- Goal test?

Backjumping (BJ)

- Similar to BT, but
  more efficient when no consistent instantiation can be found for the current var
- Instead of backtracking to most recent var...
  BJ reverts to deepest var which was c-checked against the current var

BJ Discovers
\( \{(2, 5, 3, 6)\} \) inconsistent with \( x_6 \)
No sense trying other values of \( x_6 \)

Chronological Backtracking (BT)

(e.g., depth first search)

Consistency check performed in the order in which vars were instantiated
If c-check fails,
try next value of current var
If no more values,
backtrack to most recent var
**Conflict-Directed Backjumping (CBJ)**

- More sophisticated backjumping behavior
- Each variable has conflict set $CS$
  - Set of vars that failed $c$-checks w/ current val
  - Update this set on every failed $c$-check
- When no more values to try for $x_i$
  - Backtrack to deepest var, $x_d$, in $CS(x_i)$
  - Update $CS(x_d) = CS(x_d) \cup CS(x_i) - \{x_d\}$

<table>
<thead>
<tr>
<th>CBJ Discoverer</th>
<th>$CS(x_d)$</th>
<th>$CS(x_i)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>(2, 5, 3)</td>
<td>1, 2, 3</td>
<td>1, 2, 3, 5</td>
</tr>
<tr>
<td>inconsistent with $(x_5, x_6)$</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Forward Checking (FC)**

- Perform Consistency Check Forward
- Whenever a var is assigned a value
  - Prune inconsistent values from
  - As-yet unvisited variables
  - Backtrack if domain of any var ever collapses

**Fig. 2: A fragment of the BT backtrack tree for the 8-queens problem.**

**FC only visits consistent nodes**

<table>
<thead>
<tr>
<th>but not all such nodes</th>
</tr>
</thead>
<tbody>
<tr>
<td>skips (2, 5, 3, 4) which CBJ visits</td>
</tr>
<tr>
<td>But FC can’t detect that</td>
</tr>
<tr>
<td>(2, 5, 3) inconsistent with $(x_5, x_6)$</td>
</tr>
</tbody>
</table>